

**This Page is Inserted by IFW Indexing and Scanning
Operations and is not part of the Official Record**

BEST AVAILABLE IMAGES

Defective images within this document are accurate representations of the original documents submitted by the applicant.

Defects in the images include but are not limited to the items checked:

- ☐ **BLACK BORDERS**
- ☐ **IMAGE CUT OFF AT TOP, BOTTOM OR SIDES**
- ☐ **FADED TEXT OR DRAWING**
- ☐ **BLURRED OR ILLEGIBLE TEXT OR DRAWING**
- ☐ **SKEWED/SLANTED IMAGES**
- ☐ **COLOR OR BLACK AND WHITE PHOTOGRAPHS**
- ☐ **GRAY SCALE DOCUMENTS**
- ☐ **LINES OR MARKS ON ORIGINAL DOCUMENT**
- ☐ **REFERENCE(S) OR EXHIBIT(S) SUBMITTED ARE POOR QUALITY**
- ☐ **OTHER:** _____

IMAGES ARE BEST AVAILABLE COPY.

As rescanning these documents will not correct the image problems checked, please do not report these problems to the IFW Image Problem Mailbox.

	Type	L #	Hits	Search Text	DBs	Time Stamp	Comments
1	BRS	L1	387	712/235	USPAT; US-PGP UB; EPO; JPO; DERWE NT; IBM_TD B	2004/07/29 07:44	
2	IS&R	L2	0	("cacheadjcoherency").PN.	USPAT; US-PGP UB; EPO; JPO; DERWE NT; IBM_TD B	2004/07/29 07:46	
3	BRS	L3	3105	cache adj coherency	USPAT; US-PGP UB; EPO; JPO; DERWE NT; IBM_TD B	2004/07/29 07:47	
4	BRS	L4	320	3 and speculat\$3 and indica\$3	USPAT; US-PGP UB; EPO; JPO; DERWE NT; IBM_TD B	2004/07/29 07:59	
5	BRS	L5	316	4 and processor\$3	USPAT; US-PGP UB; EPO; JPO; DERWE NT; IBM_TD B	2004/07/29 08:06	
6	BRS	L6	72	4 and page adj table	USPAT; US-PGP UB; EPO; JPO; DERWE NT; IBM_TD B	2004/07/29 07:50	
7	BRS	L7	17	6 and (speculat\$3 WITH indica\$3)	USPAT; US-PGP UB; EPO; JPO; DERWE NT; IBM_TD B	2004/07/29 08:05	

	Type	L #	Hits	Search Text	DBs	Time Stamp	Comments
8	BRS	L8	1	6 and (speculat\$3 adj execution WITH indica\$3)	USPAT; US-PGP UB; EPO; JPO; DERWE NT; IBM_TD B	2004/07/29 08:06	
9	BRS	L9	43	(speculat\$3 adj execution WITH indica\$3)	USPAT; US-PGP UB; EPO; JPO; DERWE NT; IBM_TD B	2004/07/29 08:06	
10	BRS	L10	41	9 and processor\$3	USPAT; US-PGP UB; EPO; JPO; DERWE NT; IBM_TD B	2004/07/29 08:14	
11	BRS	L11	40	10 not 6	USPAT; US-PGP UB; EPO; JPO; DERWE NT; IBM_TD B	2004/07/29 08:06	
12	BRS	L12	4	11 and MMU	USPAT; US-PGP UB; EPO; JPO; DERWE NT; IBM_TD B	2004/07/29 08:07	
13	BRS	L13	177	"19" and (memory near3 access WITH speculat\$3)	USPAT; US-PGP UB; EPO; JPO; DERWE NT; IBM_TD B	2004/07/29 08:15	
14	BRS	L14	5	10 and (memory near3 access WITH speculat\$3)	USPAT; US-PGP UB; EPO; JPO; DERWE NT; IBM_TD B	2004/07/29 11:29	

	Type	L #	Hits	Search Text	DBs	Time Stamp	Comments
15	IS&R	L15	2100	((712/3234) or (345/501,503) or (709/312) or (712/20-23,28)).CCLS.	USPAT; US-PGP UB; EPO; JPO; DERWE NT; IBM_TD B	2004/07/29 11:32	
16	BRS	L16	321	tightly near3 coupled near3 multiprocessor\$3	USPAT; US-PGP UB; EPO; JPO; DERWE NT; IBM_TD B	2004/07/29 11:33	
17	BRS	L17	17638	shared adj memory or cache adj coherency	USPAT; US-PGP UB; EPO; JPO; DERWE NT; IBM_TD B	2004/07/29 11:34	
18	BRS	L18	11	15 and 16 and 17	USPAT; US-PGP UB; EPO; JPO; DERWE NT; IBM_TD B	2004/07/29 11:34	
19	BRS	L19	0	18 and speculat\$3 near3 indicat\$3	USPAT; US-PGP UB; EPO; JPO; DERWE NT; IBM_TD B	2004/07/29 11:35	
20	BRS	L20	2	18 and speculat\$3	USPAT; US-PGP UB; EPO; JPO; DERWE NT; IBM_TD B	2004/07/29 11:50	
21	BRS	L21	9	18 not 20	USPAT; US-PGP UB; EPO; JPO; DERWE NT; IBM_TD B	2004/07/29 11:51	

	Type	L #	Hits	Search Text	DBs	Time Stamp	Comments
1	BRS	L93	1244345	control4\$ NEAR3 speculat3\$ NEARs execut3\$	USPAT; US-PGP UB; EPO; JPO; DERWE NT; IBM_TD B	2004/07/28 16:24	
2	BRS	L94	0	93 and instruction and (detect3\$ NEAR2 memory NEAR2 access2\$)	USPAT; US-PGP UB; EPO; JPO; DERWE NT; IBM_TD B	2004/07/28 16:14	
3	BRS	L95	2062285	control\$4 NEAR3 speculat\$3 NEARs execut\$3	USPAT; US-PGP UB; EPO; JPO; DERWE NT; IBM_TD B	2004/07/28 16:13	
4	BRS	L96	508	95 and instruction and (detect\$3 NEAR2 memory NEAR2 access\$2)	USPAT; US-PGP UB; EPO; JPO; DERWE NT; IBM_TD B	2004/07/28 16:25	
5	BRS	L97	0	96 and processor1\$ and (speculat\$4 adj indicat\$3)	USPAT; US-PGP UB; EPO; JPO; DERWE NT; IBM_TD B	2004/07/28 16:16	
6	BRS	L98	0	96 and processor1\$	USPAT; US-PGP UB; EPO; JPO; DERWE NT; IBM_TD B	2004/07/28 16:16	
7	BRS	L99	1	96 and (speculat\$4 adj indicat\$3)	USPAT; US-PGP UB; EPO; JPO; DERWE NT; IBM_TD B	2004/07/28 16:17	

	Type	L #	Hits	Search Text	DBs	Time Stamp	Comments
8	BRS	L100	6	96 and (speculat\$4 near3 indicat\$4)	USPAT; US-PGP UB; EPO; JPO; DERWE NT; IBM_TD B	2004/07/28 16:25	
9	IS&R	L101	7	((("20030120902") or ("5751985") or ("20030208673") or ("6684398"))).PN.	USPAT; US-PGP UB; EPO; JPO; DERWE NT; IBM_TD B	2004/07/28 16:23	
10	BRS	L102	124434 5	101 and control4\$ NEAR3 speculat3\$ NEARs execut3\$	USPAT; US-PGP UB; EPO; JPO; DERWE NT; IBM_TD B	2004/07/28 16:24	
11	BRS	L103	0	101 and (control4\$ NEAR3 speculat3\$ NEARs execut3\$)	USPAT; US-PGP UB; EPO; JPO; DERWE NT; IBM_TD B	2004/07/28 16:24	
12	BRS	L105	0	104 and instruction and (detect\$3 NEAR2 memory NEAR2 access\$2)	USPAT; US-PGP UB; EPO; JPO; DERWE NT; IBM_TD B	2004/07/28 16:30	
13	BRS	L106	0	104 and instruction and (speculat\$3 WITH memory NEAR2 access\$2)	USPAT; US-PGP UB; EPO; JPO; DERWE NT; IBM_TD B	2004/07/28 16:26	
14	BRS	L107	246	104 (speculat\$3 WITH memory NEAR2 access\$2)	USPAT; US-PGP UB; EPO; JPO; DERWE NT; IBM_TD B	2004/07/28 16:27	

	Type	L #	Hits	Search Text	DBs	Time Stamp	Comments
15	BRS	L108	0	104 and (speculat\$3 WITH memory NEAR2 access\$2)	USPAT; US-PGP UB; EPO; JPO; DERWE NT; IBM_TD B	2004/07/28 16:30	
16	BRS	L109	2	104 and (memory NEAR2 access\$2)	USPAT; US-PGP UB; EPO; JPO; DERWE NT; IBM_TD B	2004/07/28 16:30	
17	BRS	L104	3	101 and (speculat\$4 near3 indicat\$4)	USPAT; US-PGP UB; EPO; JPO; DERWE NT; IBM_TD B	2004/07/28 16:29	
18	BRS	L110	1	101 and (speculat\$3 WITH memory NEAR2 access\$2)	USPAT; US-PGP UB; EPO; JPO; DERWE NT; IBM_TD B	2004/07/28 16:30	
19	BRS	L111	0	101 and instruction and (detect\$3 NEAR2 memory NEAR2 access\$2)	USPAT; US-PGP UB; EPO; JPO; DERWE NT; IBM_TD B	2004/07/28 16:30	
20	BRS	L112	5	101 and (memory NEAR2 access\$2)	USPAT; US-PGP UB; EPO; JPO; DERWE NT; IBM_TD B	2004/07/28 16:32	
21	BRS	L113	1	112 and 110	USPAT; US-PGP UB; EPO; JPO; DERWE NT; IBM_TD B	2004/07/28 16:31	

	Type	L #	Hits	Search Text	DBs	Time Stamp	Comments
22	BRS	L114	4	112 not 113	USPAT; US-PGP UB; EPO; JPO; DERWE NT; IBM_TD B	2004/07/28 16:32	



US Patent & Trademark Office

[Subscribe \(Full Service\)](#) [Register \(Limited Service, Free\)](#) [Login](#)

 Search: ☒ The ACM Digital Library ☐ The Guide

speculation and indication and instruction and processor and 'r



THE ACM DIGITAL LIBRARY

[Feedback](#) [Report a problem](#) [Satisfaction survey](#)

Terms used

speculation and indication and instruction and processor and 'memory access' and 'page table' and 'speculative execut
Sort results by Display results
☒ [Save results to a Binder](#)
☒ [Search Tips](#)
☐ [Open results in a new window](#)
[Try an Advanced Search](#)
[Try this search in The ACM Guide](#)

Results 1 - 20 of 200

Result page: [1](#) [2](#) [3](#) [4](#) [5](#) [6](#) [7](#) [8](#) [9](#) [10](#) [next](#)

Best 200 shown

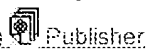
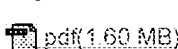
Relevance scale ☐

1 [Integrated predicated and speculative execution in the IMPACT EPIC architecture](#)

David I. August, Daniel A. Connors, Scott A. Mahlke, John W. Sias, Kevin M. Crozier, Ben-Chung Cheng, Patrick Eaton, Qudus B. Olaniran, Wen-mei W. Hwu

 April 1998 **ACM SIGARCH Computer Architecture News , Proceedings of the 25th annual international symposium on Computer architecture**, Volume 26 Issue 3

Full text available:


 Additional Information: [full citation](#), [abstract](#), [references](#), [citations](#), [index terms](#)
[Site](#)

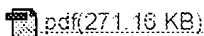
Explicitly Parallel Instruction Computing (EPIC) architectures require the compiler to express program instruction level parallelism directly to the hardware. EPIC techniques which enable the compiler to represent control speculation, data dependence speculation, and predication have individually been shown to be very effective. However, these techniques have not been studied in combination with each other. This paper presents the IMPACT EPIC Architecture to address the issues involved in design ...

2 [Effect of node size on the performance of cache-conscious B⁺-trees](#)

Richard A. Hankins, Jignesh M. Patel

 June 2003 **ACM SIGMETRICS Performance Evaluation Review , Proceedings of the 2003 ACM SIGMETRICS international conference on Measurement and modeling of computer systems**, Volume 31 Issue 2

Full text available:


 Additional Information: [full citation](#), [abstract](#), [references](#), [index terms](#)

In main-memory databases, the number of processor cache misses has a critical impact on the performance of a system. Cache-conscious indices are designed to improve performance by reducing the number of processor cache misses that are incurred during a search operation. Conventional wisdom suggests that the index's node size should be equal to the cache line size in order to minimize the number of cache misses and improve performance. As shown in this paper, this design choice ignores additional factors ...

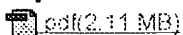
Keywords: B⁺-tree, cache-conscious, index

3 [DataScalar architectures](#)

Doug Burger, Stefanos Kaxiras, James R. Goodman

 May 1997 **ACM SIGARCH Computer Architecture News , Proceedings of the 24th annual international symposium on Computer architecture**, Volume 25 Issue 2

Full text available:


 Additional Information: [full citation](#), [abstract](#), [references](#), [citations](#), [index terms](#)

DataScalar architectures improve memory system performance by running computation redundantly across multiple processors, which are each tightly coupled with an associated memory. The program data set (and/or text) is distributed across these memories. In this execution model, each processor broadcasts operands it loads from local memory to all other units. In this paper, we describe the benefits, costs, and problems associated with the DataScalar model. We also present simulation results of ...

4 [High-bandwidth address translation for multiple-issue processors](#)

Todd M. Austin, Gurindar S. Sohi

May 1996 **ACM SIGARCH Computer Architecture News , Proceedings of the 23rd annual international symposium on Computer architecture**, Volume 24 Issue 2

Full text available: [pdf\(1.56 MB\)](#)

Additional Information: [full citation](#), [abstract](#), [references](#), [citations](#), [index terms](#)

In an effort to push the envelope of system performance, microprocessor designs are continually exploiting high levels of instruction-level parallelism, resulting in increasing bandwidth demands on the address translation mechanism. Most current microprocessor designs meet this demand with a multi-ported TLB. While this design provides an excellent hit rate at each port, its access latency and area grow very quickly as the number of ports increased. As bandwidth demands continue to increase ...

5 [Binary translation and architecture convergence issues for IBM system/390](#)

Michael Gschwind, Kemal Ebcioglu, Erik Altman, Sumedh Sathaye

May 2000 **Proceedings of the 14th international conference on Supercomputing**

Full text available: [pdf\(1.44 MB\)](#)

Additional Information: [full citation](#), [abstract](#), [references](#), [index terms](#)

We describe the design issues in an implementation of the ESA/390 architecture based on binary translation to a very long instruction word (VLIW) processor. During binary translation, complex ESA/390 instructions are decomposed into instruction "primitives" which are then scheduled onto a wide-issue machine. The aim is to achieve high instruction level parallelism due to the increased scheduling and optimization opportunities which can be exploited by binary translation software ...

6 [A dynamic scheduling logic for exploiting multiple functional units in single chip multithreaded architecture](#)

Prasad N. Golla, Eric C. Lin

February 1999 **Proceedings of the 1999 ACM symposium on Applied computing**

Full text available: [pdf\(1.19 MB\)](#)

Additional Information: [full citation](#), [references](#), [index terms](#)

Keywords: Tomasulo's algorithm, computer architecture, microprocessor, multithreading, threaded architecture

7 [Instruction cache fetch policies for speculative execution](#)

Dennis Lee, Jean-Loup Baer, Brad Calder, Dirk Grunwald

May 1995 **ACM SIGARCH Computer Architecture News , Proceedings of the 22nd annual international symposium on Computer architecture**, Volume 23 Issue 2

Full text available: [pdf\(1.14 MB\)](#)

Additional Information: [full citation](#), [abstract](#), [references](#), [citations](#), [index terms](#)

Current trends in processor design are pointing to deeper and wider pipelines and superscalar architectures. The efficient use of these resources requires *speculative execution*, a technique whereby the processor continues executing the predicted path of a branch before the branch condition is resolved. In this paper, we investigate the implications of speculative execution on instruction cache performance. We explore policies for managing instruction cache misses ranging from aggressive to conservative ...

8 [Compiler-based I/O prefetching for out-of-core applications](#)

Angela Demke Brown, Todd C. Mowry, Orran Krieger

May 2001 **ACM Transactions on Computer Systems (TOCS)**, Volume 19 Issue 2

Full text available: [pdf\(499.03 KB\)](#)

Additional Information: [full citation](#), [abstract](#), [references](#), [citations](#), [index terms](#), [review](#)

Current operating systems offer poor performance when a numeric application's working set does not fit in main memory. As a result, programmers who wish to solve "out-of-core" problems efficiently are typically faced with the onerous task of rewriting an application to use explicit I/O operations (e.g., read/write). In this paper, we propose and evaluate a fully automatic technique which liberates the programmer from this task, provides high performance and requires only minimal changes ...

Keywords: compiler optimization, prefetching, virtual memory

9 [Using value prediction to increase the power of speculative execution hardware](#)

Freddy Gabbay, Avi Mendelson

August 1998 **ACM Transactions on Computer Systems (TOCS)**, Volume 16 Issue 3

Full text available:  [pdf\(289.77 KB\)](#)

Additional Information: [full citation](#), [abstract](#), [references](#), [citations](#), [index terms](#)


This article presents an experimental and analytical study of value prediction and its impact on speculative execution in superscalar microprocessors. Value prediction is a new paradigm that suggests predicting outcome values of operations (at run-time) and using these predicted values to trigger the execution of true-data-dependent operations speculatively. As a result, stalls to memory locations can be reduced and the amount of instruction parallelism can be extended beyond the limit ...

Keywords: speculative execution, stride value prediction, value prediction

10 [Efficient and flexible value sampling](#)

M. Burrows, U. Erlingsson, S.-T. A. Leung, M. T. Vandevoorde, C. A. Waldspurger, K. Walker, W. E. Weihl

November 2000 **Proceedings of the ninth international conference on Architectural support for programming languages and operating systems**, Volume 34, 28 Issue 5, 5

Full text available:  [pdf\(191.88 KB\)](#)

Additional Information: [full citation](#), [abstract](#), [references](#), [citations](#), [index terms](#)

This paper presents novel sampling-based techniques for collecting statistical profiles of register contents, data values, and other information associated with instructions, such as memory latencies. Values of interest are sampled in response to periodic interrupts. The resulting value profiles can be analyzed by programmers and optimizer to improve the performance of production uniprocessor and multiprocessor systems. Our value sampling system extends the DCPI continuous profiling infrastructure ...

11 [Efficient and flexible value sampling](#)

M. Burrows, U. Erlingsson, S.-T. A. Leung, M. T. Vandevoorde, C. A. Waldspurger, K. Walker, W. E. Weihl

November 2000 **ACM SIGPLAN Notices**, Volume 35 Issue 11

Full text available:  [pdf\(973.26 KB\)](#)

Additional Information: [full citation](#), [abstract](#), [references](#), [citations](#), [index terms](#)

This paper presents novel sampling-based techniques for collecting statistical profiles of register contents, data values, and other information associated with instructions, such as memory latencies. Values of interest are sampled in response to periodic interrupts. The resulting value profiles can be analyzed by programmers and optimizer to improve the performance of production uniprocessor and multiprocessor systems. Our value sampling system extends the DCPI continuous profiling infrastructure ...

12 [A scalable cross-platform infrastructure for application performance tuning using hardware counters](#)

S. Browne, J. Dongarra, N. Garner, K. London, P. Mucci

November 2000 **Proceedings of the 2000 ACM/IEEE conference on Supercomputing (CDROM)**

Full text available:

 [pdf\(2.82 MB\)](#)  [Publisher Site](#)


Additional Information: [full citation](#), [abstract](#), [references](#), [citations](#), [index terms](#)

The purpose of the PAPI project is to specify a standard API for accessing hardware performance counters available on most modern microprocessors. These counters exist as a small set of registers that count "events", which are occurrences of specific signals and states related to the processor's function. Monitoring these events facilitates correlation between the structure of source/object code and the efficiency of the mapping of that code to the underlying architecture. This ...

13 [Speculative execution exception recovery using write-back suppression](#)

Roger A. Bringmann, Scott A. Mahlke, Richard E. Hank, John C. Gyllenhaal, Wen-mei W. Hwu

December 1993 **Proceedings of the 26th annual international symposium on Microarchitecture**

Full text available:  [pdf\(1.22 MB\)](#)

Additional Information: [full citation](#), [references](#), [citations](#)

Keywords: VLIW, exception detection, exception recovery, scheduling, speculative execution, superscalar

14 [Speculative execution: Enhancing memory level parallelism via recovery-free value prediction](#)

Huiyang Zhou, Thomas M. Conte

June 2003 **Proceedings of the 17th annual international conference on Supercomputing**

Full text available:  [pdf\(302.33 KB\)](#)

Additional Information: [full citation](#), [abstract](#), [references](#), [index terms](#)


The ever-increasing computational power of contemporary microprocessors reduces the execution time spent on arithmetic computations (i.e., the computations not involving slow memory operations such as cache misses) significantly. Therefore, for memory intensive workloads, it becomes more important to overlap multiple cache misses than to overlap slow memory operations with other computations. In this paper, we propose a novel technique to parallelize sequential cache misses, thereby increasing m ...

Keywords: memory disambiguation, memory level parallelism, prefetching, recovery-free value prediction

15 [DAISY: dynamic compilation for 100% architectural compatibility](#)

Kemal Ebcioglu, Erik R. Altman

May 1997 **ACM SIGARCH Computer Architecture News , Proceedings of the 24th annual international symposium on Computer architecture**, Volume 25 Issue 2

Full text available:  [pdf\(1.97 MB\)](#)



Additional Information: [full citation](#), [abstract](#), [references](#), [citations](#), [index terms](#)

Although VLIW architectures offer the advantages of simplicity of design and high issue rates, a major impediment to their use is that they are not compatible with the existing software base. We describe new simple hardware for a VLIW machine we call **DAISY** (*Dynamically Architected Instruction Set from Yorktown*). **DAISY** is specifically intended to emulate existing architectures, so that all existing software ...

Keywords: binary translation, dynamic compilation, instruction-level parallelism, object code compatible VLIW superscalar

16 [Computing curricula 2001](#)

September 2001 **Journal on Educational Resources in Computing (JERIC)**



Full text available:  [pdf\(613.63 KB\)](#)  [html](#)
(2.78 KB)

Additional Information: [full citation](#), [references](#), [citations](#), [index terms](#)

17 [Avoiding initialization misses to the heap](#)

Jarrold A. Lewis, Bryan Black, Mikko H. Lipasti

May 2002 **ACM SIGARCH Computer Architecture News**, Volume 30 Issue 2

Full text available:  [pdf\(1.29 MB\)](#)  [Publisher Site](#)

Additional Information: [full citation](#), [abstract](#), [references](#), [index terms](#)


This paper investigates a class of main memory accesses (*invalid memory traffic*) that can be eliminated altogether. Invalid memory traffic is real data traffic that transfers invalid data. By tracking the initialization of dynamic memory allocations, it is possible to identify store instructions that miss the cache and would fetch uninitialized heap data. The data transfers associated with these initialization misses can be avoided without losing correctness. The memory system property can ...

Keywords: invalid memory traffic, initializing stores, cache installation, allocation range cache

18 [Speeding up irregular applications in shared-memory multiprocessors: memory binding and group prefetching](#)

Zheng Zhang, Josep Torrellas

May 1995 **ACM SIGARCH Computer Architecture News , Proceedings of the 22nd annual international symposium on Computer architecture**, Volume 23 Issue 2

Full text available:  [pdf\(1.74 MB\)](#)

Additional Information: [full citation](#), [abstract](#), [references](#), [citations](#), [index terms](#)


While many parallel applications exhibit good spatial locality, other important codes in areas like graph problem solving or CAD do not. Often, these irregular codes contain small records accessed via pointers. Consequently the former applications benefit from long cache lines, the latter prefer short lines. One good solution is to combine short lines with prefetching. In this way, each application can exploit the amount of spatial locality that it has

However, prefetching, if provided, ...

19 [Data speculation support for a chip multiprocessor](#)

Lance Hammond, Mark Willey, Kunle Olukotun

October 1998 **Proceedings of the eighth international conference on Architectural support for programming languages and operating systems**, Volume 32 , 33 Issue 5 , 11

Full text available:  [pdf\(1.75 MB\)](#)

Additional Information: [full citation](#), [abstract](#), [references](#), [citations](#), [index terms](#)



Thread-level speculation is a technique that enables parallel execution of sequential applications on a multiprocessor. This paper describes the complete implementation of the support for threadlevel speculation on the Hydra chip multiprocessor (CMP). The support consists of a number of software speculation control handlers and modifications to the shared secondary cache memory system of the CMP. This support is evaluated using five representative integer applications. Our results show that the s ...

20 [Tuning compiler optimizations for simultaneous multithreading](#)

Jack L. Lo, Susan J. Eggers, Henry M. Levy, Sujay S. Parekh, Dean M. Tullsen

December 1997 **Proceedings of the 30th annual ACM/IEEE international symposium on Microarchitecture**

Full text available:

 [pdf\(1.45 MB\)](#)  [Publisher](#)
[Site](#)

Additional Information: [full citation](#), [abstract](#), [references](#), [citations](#), [index terms](#)

Compiler optimizations are often driven by specific assumptions about the underlying architecture and implementation of the target machine. For example, when targeting shared-memory multiprocessors, parallel programs are compiled to minimize sharing, in order to decrease high-cost, inter-processor communication. This paper reexamines several compiler optimizations in the context of simultaneous multithreading (SMT), a processor architecture that issues instructions from multiple threads to the functional units ...

Keywords: cache size, compiler optimizations, cyclic algorithm, fine-grained sharing, instructions, inter-processor communication, inter-thread instruction-level parallelism, latency hiding, loop tiling, loop-iteration scheduling, memory system resources, optimising compilers, parallel architecture, parallel programs, performance, processor architecture, shared-memory multiprocessors, simultaneous multithreading, software speculative execution

Results 1 - 20 of 200

Result page: [1](#) [2](#) [3](#) [4](#) [5](#) [6](#) [7](#) [8](#) [9](#) [10](#) [next](#)

The ACM Portal is published by the Association for Computing Machinery. Copyright © 2004 ACM, Inc.

[Terms of Usage](#) [Privacy Policy](#) [Code of Ethics](#) [Contact Us](#)

Useful downloads:  [Adobe Acrobat](#)  [QuickTime](#)  [Windows Media Player](#)  [Real Player](#)